PATENT
Attorney Docket No. 200352

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

PARKES et al.

Group Art Unit: 2127

Application No. 09/436,618

Examiner: Ali, Syed J.

Filed: November 9, 1999

For: A METHOD OF PIPELINED PROCESSING OF PROGRAM DATA

PENDING CLAIMS AFTER AMENDMENTS MADE IN RESPONSE TO OFFICE ACTION DATED FEBUARY 12, 2003

1. A method of carrying out a procedure on a computer system having a memory, the memory containing user context data and global data, comprising: executing a first server, wherein the first server defines a computer-executable function for performing a first sub-task of the procedure; manipulating the global data to carry out the first sub-task; sending the user context data to a second server; executing the second server, wherein the second server defines a computer-executable function for performing a second sub-task of the procedure; and manipulating the global data to carry out the second sub-task using the user context data, wherein the first and second servers are optimized to execute in cache.

2. The method of claim 1, further comprising storing the user context data in a work packet and sending the work packet from the first server to the second server, wherein the work packet contains an action code for describing an action to be performed by the second server.

3. The method of claim 2, wherein the work packet contains a reply state, and the method further comprises: causing the second server to update the work packet by replacing a value contained in the action code with a value contained in the reply state; and causing the second server to send the updated work packet back to the first server.

4. The method of claim 1, further comprising: in response to receiving a first work packet containing the user context data; causing the first server to partly perform the first sub-task using the first work packet; sending a second work packet containing the user context data from the first server to the second server; causing the second server to perform the second sub-task using the second work packet and store a result of the second sub-task in the second work packet; and sending the second work packet from the second server to the first server, wherein the result is useable by the first server to complete the performance of the first sub-task.

5. The method of claim 4, wherein the second work packet is linked as a child to the first work packet.

6. The method of claim 1, wherein the computer system has a first CPU and a second CPU, and the cache is comprised of a first area usable by the first CPU and a second are usable by the second CPU, and the executable code of the first server is optimized to fit in the first area and the executable code of the second server is optimized to fit in the second area.

7. The method of claim 1, wherein the procedure is a search of a database index tree containing a plurality of nodes, the first sub-task is to examine a node and the second sub-task is to perform an input/output operation for retrieving the node from memory and storing the node in the cache.

8. The method of claim 7, further comprising: in response to receiving a first work packet containing the user context data; causing the first server to determine if a node is in the cache; and if the node is determined not to be in the cache, sending a second work packet containing the user context data from the first server to the second server; causing the second server to retrieve the node from a main memory using the second work packet and store the node in the cache; causing the second server to store a reference to the cached node in the second work

packet; and sending the second work packet from the second server to the first server, wherein the first server responds to the receipt of the second work packet by searching the cached node.

9. The method of claim 8, wherein the first work packet contains a reference to a parent work packet.

10. The method of claim 1, wherein the computer system has a plurality of CPUs, and at least one server executes on only one CPU at a time.

11. The method of claim 1, wherein the computer system has a plurality of CPUs, and at least two instances of one of the servers execute concurrently on different CPUs.

12. The method of claim 1, wherein the computer system has a first CPU and a second CPU, and the work packet has a designated value, and wherein one of the servers executes on the first CPU when the designated value falls within a first range and executes on the second CPU when the designated value falls within a second range.

13. A method of writing a computer program for carrying out a procedure on computer system having a cache, global data and a user context comprising: dividing the procedure into sub-tasks; defining a server for each sub-task, wherein each server contains instructions for performing its respective sub-task on the global data and wherein each server is optimized to fit inside the cache when executed; and defining a work packet for transferring the user context between two or more of the servers.

14. The method of claim 13, further comprising defining an action code to be located in the work packet for describing an action to be performed by a server.

15. The method of claim 13, further comprising defining a reply state code for the work packet, the reply state code being useable by a server to gain information about results of a function executed by another server.

16. The method of claim 13, further comprising: defining a first work packet for a first server; defining a second work packet for a second server, wherein the first work packet is usable by the first server to perform a first sub-task on the global data and the second work packet is usable by the second server to receive the user context from the first server, perform the second task, and return a result of the second task to the first server.

17. The method of claim 16, wherein the second work packet is linked as a child to the first work packet.

18. The method of claim 13, wherein the computer system has a first CPU and a second CPU, and the cache is comprised of a first area usable by the first CPU and a second are usable by the second CPU, and the first server is optimized to fit in the first area and the second server is optimized to fit in the second area.

19. The method of claim 13, wherein the procedure is a search of a database index tree containing a plurality of nodes, the first sub-task is to examine a node and the second sub-task is to perform an input/output operation for retrieving the node from memory.

20. The method of claim 13, wherein the computer system has a plurality of CPUs, and at least one of the servers is defined to execute on only one CPU at a time.

21. The method of claim 13, wherein the computer system has a plurality of CPUs and at least one of the servers is defined to run concurrently as at least two instances on different CPUs.

22. The method of claim 13, wherein the computer system has a first CPU and a second CPU, a designated value field is defined for the work packet, and wherein at least one of the servers executes on the first CPU when the designated value falls within a first range and executes on the second CPU when the designated value falls within a second range.

23. A computer-readable medium having stored thereon a data structure, the data structure comprising: a work packet for transferring user context information between at least two servers, wherein each server defines at least one function for performing a sub-task of a computer-executable procedure to manipulate a global data set.

24. The computer-readable medium of claim 23, wherein the work packet has defined therein an action code for describing an action to be performed by one of the servers.

25. The computer-readable medium of claim 24, wherein the work packet has defined therein a reply state usable by one of the servers to send a result of its sub-task to another server.

26. A computer-readable medium having stored thereon a data structure, the data structure comprising: a first server defining at least one function for performing a sub-task of a computer-executable procedure to manipulate a global data set, wherein the first server executes the function in response to the receipt of a first work packet, the first work packet containing user context information usable by the first server to perform the sub-task, and wherein the first server transmits the user context information to a second server using a second work packet.

27. The computer-readable medium of claim 26, wherein the second server stores a result of a second sub-task performed on the global data in the second work packet and returns the second work packet to the first server.

28. A computer-readable medium having computer-executable instructions for

performing, on a computer system having a memory, user context data and global data, the method of claim 1.

29. The computer-readable medium of claim 28, having further computer-executable instructions for: storing the user context data in a work packet and sending the work packet from the first server to the second server, wherein the work packet contains an action code for describing an action to be performed by the second server.

30. The computer-readable medium of claim 29, wherein the work packet contains a reply state, and the computer-readable medium has further computer-executable instructions for: causing the second server to update the work packet by replacing a value contained in the action code with a value contained in the reply state; and causing the second server to send the updated work packet back to the first server.

31. The computer-readable medium of claim 28, having further computer-executable instructions for: in response to receiving a first work packet containing the user context data; causing the first server to partially complete the first sub-task using the first work packet; sending a second work packet containing the user context data from the first server to the second server; causing the second server to perform the second sub-task using the second work packet and store a result of the second sub-task in the second work packet; sending the second work packet to the first server, wherein the result is useable by the first server to fully complete of the first sub-task.

32. The computer-readable medium of claim 28, wherein the computer system has a first CPU and a second CPU, and the cache is comprised of a first area usable by the first CPU and a second are usable by the second CPU, and the executable code of the first server is optimized to fit in the first area and the executable code of the second server is optimized to fit in the second area.

33. The computer-readable medium of claim 28, wherein the procedure is a search of a database index tree containing a plurality of nodes, the first sub-task is to examine a node and the second sub-task is to perform an input/output operation for retrieving the node from memory and storing the node in the cache.

34. The computer-readable medium of claim 33 having further computer-executable instructions for: in response to receiving a first work packet containing the user context data; causing the first server to determine if a node is in the cache; and if the node is determined not to be in the cache, sending a second work packet containing the user context data from the first server to the second server; causing the second server to retrieve the node from a main memory using the second work packet and store the node in the cache; causing the second server to store a reference to the cached node in the second work packet; and sending the second work packet from the second server to the first server, wherein the first server searches the cached node.

35. The computer-readable medium of claim 34, wherein the first work packet contains a reference to a parent work packet.

36. The computer-readable medium of claim 28, wherein the computer system has a plurality of CPUs, and at least one server executes on only one CPU at a time.

37. The computer-readable medium of claim 28, wherein the computer system has a plurality of CPUs, and at least two instances of one of the servers execute concurrently on different CPUs.

38. The computer-readable medium of claim 28, wherein the computer system has a first CPU and a second CPU, and the work packet has a designated value, and wherein one of the servers executes on the first CPU when the designated value falls within a first range and executes on the second CPU when the designated value falls within a second range.